AMENDMENTS TO THE CLAIMS

Please enter the following amendments:

1. (Currently Amended) A method for processing data using a programmable processor comprising:

decoding a single instruction for writing data to memory based on a mask and data contained in at least one register, the mask consisting of N independently selectable mask bits, N being an integer multiple of eight, each of the mask bits corresponding to a data bit contained in the at least one register ~~comprising a plurality of mask fields that each corresponds to a data field of the data contained in the at least one register~~, each of the ~~plurality of mask fields~~ mask bits being independently selectable as either a write-enabled ~~mask field~~ mask bit or a write-disabled ~~mask field~~ mask bit;

detecting some of the ~~mask fields~~ mask bits of the mask as being selected as write-enabled ~~mask fields~~ mask bits to identify corresponding ~~data fields~~ data bits of the data contained in the at least one register as write-enabled ~~data fields~~ data bits; and

writing the write-enabled ~~data fields~~ data bits to a specified memory location.

2. (Canceled)

3. (Canceled)

4. (Currently Amended) The method of claim 1 wherein the writing step further comprises reading an unaltered field of data from the specified memory location and writing the unaltered field of data along with the write-enabled ~~data fields~~ data bits to the specified memory location.

5. (Original) The method of claim 1 wherein the mask is contained in a specified register.

6. (Previously Presented) The method of claim 1 wherein the memory location is specified by a register.

7. (Original) The method of claim 1 wherein the specified memory location comprises a section of memory having a specific width and beginning at a specific memory address.

8. (Currently Amended) The method of claim 1 wherein each write-enabled ~~mask field~~ mask bit is indicated as a logic 1.

9. (Previously Presented) The method of claim 1 further comprising:

decoding a second single instruction specifying a register containing a first plurality of floating-point operands and another register containing a second plurality of floating-point operands;

multiplying the first plurality of floating-point operands by the second plurality of floating-point operands to produce a plurality of products;

and providing the plurality of products to partitioned fields of a result register as a catenated result.

10. (Currently Amended) A computer-readable storage medium having stored therein a plurality of instructions that cause a programmable processor to perform data operations:

at least some of the instructions including a single instruction for selectively storing data, the single instruction capable of instructing the programmable processor to perform operations comprising:

decoding the single instruction to obtain a mask and data contained in at least one register, the mask consisting of N independently selectable mask bits, N being an integer multiple of eight, each of the mask bits corresponding to a data bit contained in the at least one register ~~comprising a plurality of mask fields that each corresponds to a data field of the data contained in the at least one register~~, each of the ~~plurality of mask fields~~ mask bits being independently selectable as either a write-enabled ~~mask field~~ mask bit or a write-disabled ~~mask field~~ mask bit;

detecting some of the ~~fields~~ mask bits of the mask as being selected as write-enabled ~~mask fields~~ mask bits to identify corresponding ~~data fields~~ data bits of the data contained in the at least one register as write-enabled ~~data fields~~ data bits; and

writing the write-enabled ~~data fields~~ data bits to a specified memory location.

11. (Canceled)

12. (Canceled)

13. (Currently Amended) The computer-readable storage medium of claim 10 wherein the writing step further comprises reading an unaltered field of data from the specified memory location and writing the unaltered field of data along with the write-enabled ~~data fields~~ data bits to the specified memory location.

14. (Previously Presented) The computer-readable storage medium of claim 10 wherein the mask is contained in a specified register.

15. (Previously Presented) The computer-readable storage medium of claim 10 wherein the memory location is specified by a register.

16. (Previously Presented) The computer-readable storage medium of claim 10 wherein the specified memory location comprises a section of memory having a specific width and beginning at a specific memory address.

17. (Currently Amended) The computer-readable storage medium of claim 10 wherein each write-enabled ~~mask field~~ mask bit is indicated as a logic 1.

18. (Previously Presented) The computer-readable storage medium of claim 10 wherein at least some of the instructions further include a group floating-point multiply instruction for multiplying floating-point data in the programmable processor, the group floating-point multiply instruction capable of instructing the programmable processor to perform operations comprising:

decoding the group floating-point multiply instruction specifying a register containing a first plurality of floating-point operands and another register containing a second plurality of floating-point operands;

multiplying the first plurality of floating-point operands by the second plurality of floating-point operands to produce a plurality of products; and

providing the plurality of products to partitioned fields of a result register as a catenated result.

19 - 27. (Canceled)

28. (Currently Amended) A method for processing data in a programmable processor, the method comprising:

decoding a single instruction for performing a bitwise insert operation on data in at least one register in a register file within the programmable processor, the bitwise insert operation operating on a first operand and a second operand stored in the at least one register in the register file, the second operand consisting of N independently selectable bits, N being an integer multiple of eight, wherein each bit in the second operand is ~~individually~~ independently selectable as either having a first predetermined value or a second predetermined value; and

for each bit in the first operand, the bitwise insert operation inserting the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has ~~a~~ the first predetermined value.

29. (Previously Presented) The method of claim 28 wherein the first predetermined value is a logic 1.

30. (Previously Presented) The method of claim 28 wherein for each bit in the first operand, a corresponding bit position in the destination value is maintained as unchanged if a corresponding bit in the second operand has a the second predetermined value.

31. (Previously Presented) The method of claim 30 wherein the second predetermined value is a logic 0.

32. (Previously Presented) The method of claim 28 further comprising a step of storing the destination value into memory.

33. (Previously Presented) The method of claim 28 wherein each of the first and second operands has a width of 64 bits.

34. (Previously Presented) The method of claim 28 further comprising a step of executing a plurality of different group floating-point arithmetic operations that arithmetically operate on multiple floating-point operands stored in partitioned fields of registers in the register file to produce a catenated result that is returned to a register in the register file, wherein the catenated result comprises a plurality of individual floating-point results.

35. (Currently Amended) A computer-readable storage medium having stored therein a plurality of instructions that cause a programmable processor to perform operations on data in the programmable processor, the plurality of instructions comprising:

an instruction that causes the processor to perform a bitwise insert operation on data in at least one register in a register file within the programmable processor, the bitwise insert operation operating on a first operand and a second operand stored in the at least one register in the register file, the second operand consisting of N independently selectable bits, N being an integer multiple of eight, wherein each bit in the second operand is ~~individually~~ independently selectable as either having a first predetermined value or a second predetermined value; and

wherein for each bit in the first operand, the bitwise insert operation inserts the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has ~~a~~ the first predetermined value.

36. (Previously Presented) The computer-readable storage medium of claim 35 wherein the first predetermined value is a logic 1.

37. (Previously Presented) The computer-readable storage medium of claim 35 wherein for each bit in the first operand, a corresponding bit position in the destination value is maintained as unchanged if a corresponding bit in the second operand has the second predetermined value.

38. (Previously Presented) The computer-readable storage medium of claim 37 wherein the second predetermined value is a logic 0.

39. (Previously Presented) The computer-readable storage medium of claim 35 wherein the destination value is stored into memory.

40. (Previously Presented) The computer-readable storage medium of claim 35 wherein each of the first and second operands has a width of 64 bits.

41. (Previously Presented) The computer-readable storage medium of claim 35 wherein the plurality of instructions further comprises a plurality of different group floating-point arithmetic operations that arithmetically operate on multiple floating-point operands stored in partitioned fields of registers in the register file to produce a catenated result that is returned to a register in the register file, wherein the catenated result comprises a plurality of individual floating-point results.